Defense Technical Information Center Compilation Part Notice

ADP010514

TITLE: An Optimal Control Theory Based Algorithm to Solve 2D Aerodynamic Shape Optimisation Problems for Inviscid and Viscous Flows

This paper is part of the following report:

TITLE: Aerodynamic Design and Optimisation of Flight Vehicles in a Concurrent Multi-Disciplinary Environment [la Conception et l'optimisation aerodynamiques des vehicules aeriens dans un environnement pluridisciplinaire et simultane]

To order the complete compilation report, use: ADA388284

The following component part numbers comprise the compilation report:

ADP010499 thru ADP010530

An Optimal Control Theory Based Algorithm to solve 2D Aerodynamic Shape Optimisation Problems for Inviscid and Viscous Flows

S.Hiernaux* and J.-A. Essers†
Aerodynamics group, University of Liège
Rue Ernest Solvay 21 (Bât C3), B-4000 Liège, Belgium

March 2, 2000

1 Introduction

With the capacity of today's computers, one can envisage the resolution of shape optimization problems in aerodynamics. Nevertheless, optimization methods require many evaluations of different aerodynamic configurations, and so are much more expensive than a single analysis. It is therefore mandatory to find methods that evaluate aerodynamic functions and their gradient at the lowest possible computational cost, as well as fast and robust optimization methods.

Classical optimization techniques (descent methods) not only require the value of the function to optimize, but also of its gradient. The classical way to compute the gradient is to use a finite-difference formula; the main drawback of this method is due to the fact that $n+1$ evaluations of aerodynamic functions are necessary at each iteration, n being the number of parameters defining the geometry to optimize. So, such methods are completely unsuited to aerodynamic shape optimization, because of the high computational cost of the single analysis. Alternative methods (stochastic optimization, genetic algorithms) that don't require gradient information are also highly costly in term of CPU time.

For a few years, techniques for sensitivity analysis based on the optimal control theory have been developed ([1],[13]). These techniques derive from the state equations another set of equations called "adjoint" or "costate" equations. The solution of these adjoint equations is used to compute the gradient at very low cost; since solving the adjoint equations is equivalent to solve the state equations, the cost of sensitivity analysis is greatly reduced. Some authors use adjoint equations derived from the discretized Euler equations ([5],[7]). In this paper, we focus on adjoint equations derived from analytical state equations.

*F.N.R.S. Research Assistant
†Professor

2 Optimal control

A typical shape optimization problem can be stated as follows:

Let Ω be a subspace of $\mathbf{R}^2$. Find the shape of Γ_0, a boundary of Ω controlled by design variables $\mathbf{u}$, such that the functional:

$$\mathcal{J}(\mathbf{u}) = \int_{\Gamma_f} \Phi(\mathbf{s}, \mathbf{q}_x, \mathbf{q}_y, \mathbf{u}) d\Gamma_f \qquad (1)$$

computed on a boundary Γ_f (where Γ_f and Γ_0 can be different) is minimized

Over the domain Ω, the state variables $\mathbf{s}$ are governed by the advective-diffusive state equations:

$$\partial_x \left[\mathbf{f}(\mathbf{s}) - \mathbf{f}_v(\mathbf{q}, \mathbf{q}_x, \mathbf{q}_y)\right] + \partial_y \left[\mathbf{g}(\mathbf{s}) - \mathbf{g}_v(\mathbf{q}, \mathbf{q}_x, \mathbf{q}_y)\right] = 0 \quad \text{on } \Omega \; (2)$$

$$\mathbf{G}_i(\mathbf{q}, \mathbf{q}_x, \mathbf{q}_y, \mathbf{u}) = 0 \quad \text{on } \Gamma_i \; (3)$$

where $\mathbf{q} = \mathbf{q}(\mathbf{s})$ are the primitive variables and

$$\mathbf{q}_x = \partial_x \mathbf{q} = \mathbf{M} \partial_x \mathbf{s} \qquad (4)$$

$$\mathbf{q}_y = \partial_y \mathbf{q} = \mathbf{M} \partial_y \mathbf{s} \qquad (5)$$

with $\mathbf{M} = \frac{\partial \mathbf{q}}{\partial \mathbf{s}}$ the jacobian of the transformation between primitive and state variables.

Paper presented at the RTO AVT Symposium on "Aerodynamic Design and Optimisation of Flight Vehicles in a Concurrent Multi-Disciplinary Environment", held in Ottawa, Canada, 18-21 October 1999, and published in RTO MP-35.

In order to find an optimality condition, one have to form the *Lagrangian* of the problem:

$$\begin{aligned}\mathcal{L} &= \mathcal{J} \\ &+ \int_\Omega \psi^T \left[\partial_x \left(\mathbf{f}-\mathbf{f}_v\right) + \partial_y \left(\mathbf{g}-\mathbf{g}_v\right)\right] d\Omega \\ &+ \sum_i \int_{\Gamma_i} \mu_i^T \mathbf{G}_i \, d\Gamma_i \\ &+ \int_\Omega \lambda_x^T \left(\mathbf{q}_x - \mathbf{M}\partial_x \mathbf{s}\right) + \lambda_y^T \left(\mathbf{q}_x - \mathbf{M}\partial_x \mathbf{s}\right) d\Omega \end{aligned} \tag{6}$$

where $\psi, \mu_i, \psi_x, \psi_y$ are the costate or adjoint variables associated with the state equations, state boundary conditions and primitive-conservative realtions respectively.

The optimal control theory states that the Lagrangian is stationary at the optimum solution, i.e. at the solution, the (independant) variations of the Lagrangian with respect to the state ($\mathbf{s}$), costate (ψ, μ) and design ($\mathbf{u}$) variables is zero.

The variation of the Lagrangian due to the design variables is the sum of two contributions:

- a *local variation* due to the variation of the design variables;
- a *convective variation* due to the displacement of the domain and boundaries.

The variations of the Lagrangian due to the state, and costate variables are local variations.

2.1 Local variations

The local variation of the lagrangian can be split in eight parts:

$$\begin{aligned}\delta\mathcal{L}_{\text{loc}} &= \delta\mathcal{L}^{\mathbf{s}}_{\text{loc}} + \delta\mathcal{L}^{\mathbf{q_x}}_{\text{loc}} + \delta\mathcal{L}^{\mathbf{q_y}}_{\text{loc}} + \delta\mathcal{L}^{\mathbf{u}}_{\text{loc}} + \delta\mathcal{L}^{\psi}_{\text{loc}} \\ &+ \sum_i \delta\mathcal{L}^{\mu_i}_{\text{loc}} + \delta\mathcal{L}^{\lambda_x}_{\text{loc}} + \delta\mathcal{L}^{\lambda_y}_{\text{loc}} \end{aligned} \tag{7}$$

As the variations of the state, costate and design variables are independent, the variation of their contribution to the variation of the Lagrangian must be zeroed independently.

It is easy to demonstrate that setting $\delta\mathcal{L}^{\psi}_{\text{loc}}$ and $\delta\mathcal{L}^{\mu_i}_{\text{loc}}$ to zero leads to the state equations and boundary conditions, while setting $\delta\mathcal{L}^{\lambda_x}_{\text{loc}}$ and $\delta\mathcal{L}^{\lambda_y}_{\text{loc}}$ to zero leads to the primitive-conservative relations.

The local variation due to the design variables is:

$$\delta\mathcal{L}^{\mathbf{u}}_{\text{loc}} = \sum_i \int_{\Gamma_i} \frac{\partial \mathcal{H}_i}{\partial \mathbf{u}} \delta\mathbf{u} \, d\Gamma_i \tag{8}$$

where the *Hamiltonians* $\mathcal{H}$ are defined as:

- $\mathcal{H}_f = \Phi + \mu_f^T \mathbf{G}_f$ on Γ_f;
- $\mathcal{H}_i = \mu_i^T \mathbf{G}_i$ on the other boundaries.

The local variation due to the state variable is,using Gauss' theorem:

$$\begin{aligned}\delta\mathcal{L}^{\mathbf{s}}_{\text{loc}} &= -\int_{\cup\Gamma} \left\{\psi^T \left[\left(\mathbf{A}-\mathbf{A}_v\right) n_x + \left(\mathbf{B}-\mathbf{B}_v\right) n_y\right]\right\} \delta\mathbf{s} \, d\Gamma_i \\ &- \int_{\cup\Gamma} \left\{\left[\lambda_x^T n_x + \lambda_y^T n_y\right] \mathbf{M} + \mu_i^T \partial_{\mathbf{s}} \mathbf{G}_i + \partial_{\mathbf{s}} \Phi\right\} \delta\mathbf{s} \, d\Gamma_i \\ &+ \int_\Omega \left\{\left(\mathbf{A}-\mathbf{A}_v\right)^T \partial_x \psi + \left(\mathbf{B}-\mathbf{B}_v\right)^T \partial_y \psi\right\} \delta\mathbf{s} \, d\Omega \\ &- \int_\Omega \left\{\partial_x \left(\mathbf{M}^T \lambda_x\right) - \partial_y \left(\mathbf{M}^T \lambda_y\right) + \lambda_x^T \frac{\partial \mathbf{M}}{\partial \mathbf{s}} \partial_x \mathbf{s} \right. \\ &\left. + \lambda_y^T \frac{\partial \mathbf{M}}{\partial \mathbf{s}} \partial_y \mathbf{s}\right\} \delta\mathbf{s} \, d\Omega \end{aligned} \tag{9}$$

or

$$\begin{aligned}\delta\mathcal{L}^{\mathbf{s}}_{\text{loc}} &= -\int_{\cup\Gamma} \left\{\psi^T \left[\left(\mathbf{A}-\mathbf{A}_v\right) n_x + \left(\mathbf{B}-\mathbf{B}_v\right) n_y\right]\right\} \delta\mathbf{s} \, d\Gamma_i \\ &- \int_{\cup\Gamma} \left\{\left[\lambda_x^T n_x + \lambda_y^T n_y\right] \mathbf{M} + \mu_i^T \partial_{\mathbf{s}} \mathbf{G}_i + \partial_{\mathbf{s}} \Phi\right\} \delta\mathbf{s} \, d\Gamma_i \\ &+ \int_\Omega \left\{\left(\mathbf{A}-\mathbf{A}_v\right)^T \partial_x \psi + \left(\mathbf{B}-\mathbf{B}_v\right)^T \partial_y \psi\right\} \delta\mathbf{s} \, d\Omega \\ &- \int_\Omega \left\{\mathbf{M}^T \partial_x \lambda_x - \mathbf{M}^T \partial_y \lambda_y\right\} \delta\mathbf{s} \, d\Omega \end{aligned} \tag{10}$$

where

$$\begin{aligned} \mathbf{A} &= \frac{\partial \mathbf{f}}{\partial \mathbf{s}} \quad \mathbf{B} = \frac{\partial \mathbf{g}}{\partial \mathbf{s}} \\ \mathbf{A}_v &= \frac{\partial \mathbf{f}_v}{\partial \mathbf{q}} \quad \mathbf{B}_v = \frac{\partial \mathbf{g}_v}{\partial \mathbf{q}} \end{aligned} \tag{11}$$

The local variation due to $\mathbf{q}_x$ and $\mathbf{q}_y$ are, using Gauss' theorem:

$$\begin{aligned}\delta\mathcal{L}^{\mathbf{q_x}}_{\text{loc}} &= \int_{\cup\Gamma} \left\{\partial_{\mathbf{q}_x} \Phi + \sum_i \mu_i^T \partial_{\mathbf{q}_x} \mathbf{G}_i - \psi^T \left(\mathbf{C} n_x + \mathbf{E} n_y\right)\right\} \delta\mathbf{q}_x \, d\Gamma \\ &+ \int_\Omega \left\{\lambda_x + \left(\mathbf{C}^T \partial_x \psi + \mathbf{E}^T \partial_y \psi\right)\right\} \delta\mathbf{q}_x \, d\Omega \end{aligned} \tag{12}$$

$$\delta\mathcal{L}_{\text{loc}}^{\mathbf{q_y}} = \int_{\cup\Gamma}\Big\{\partial_{\mathbf{q}_y}\Phi + \sum_i \mu_i^T\partial_{\mathbf{q}_y}\mathbf{G}_i - \psi^T\left(\mathbf{D}n_x + \mathbf{F}n_y\right)\Big\}\delta\mathbf{q}_y\, d\Gamma + \int_\Omega\left\{\lambda_y + \left(\mathbf{D}^T\partial_x\psi + \mathbf{F}^T\partial_y\psi\right)\right\}\delta\mathbf{q}_y\, d\Omega$$

where

$$\mathbf{C} = \frac{\partial \mathbf{f}_v}{\partial \mathbf{q}_x} \quad \mathbf{E} = \frac{\partial \mathbf{g}_v}{\partial \mathbf{q}_x}$$
$$\mathbf{D} = \frac{\partial \mathbf{f}_v}{\partial \mathbf{q}_y} \quad \mathbf{F} = \frac{\partial \mathbf{g}_v}{\partial \mathbf{q}_y}$$

Equaling the integrals over Ω to zero in (12) and (13) gives the relations between ψ, λ_x and λ_y:

$$\lambda_x = -\left(\mathbf{C}^T\partial_x\psi + \mathbf{E}^T\partial_y\right) \quad (14)$$
$$\lambda_y = -\left(\mathbf{D}^T\partial_x\psi + \mathbf{F}^T\partial_y\right) \quad (15)$$

Equaling the integral over Ω to zero in (10) and using the two above relations gives the *adjoint equations*:

$$\left(\mathbf{A} - \mathbf{A}_v\right)^T\partial_x\psi + \left(\mathbf{B} - \mathbf{B}_v\right)^T\partial_y\psi + \mathbf{M}^T\left(\partial_x\tilde{\mathbf{f}}_v + \partial_y\tilde{\mathbf{g}}_v\right) = 0 \quad (16)$$

with:

$$\tilde{\mathbf{f}}_v = \left(\mathbf{C}^T\partial_x\psi + \mathbf{E}^T\partial_y\psi\right)$$
$$\tilde{\mathbf{g}}_v = \left(\mathbf{D}^T\partial_x\psi + \mathbf{F}^T\partial_y\psi\right)$$

the "adjoint viscous fluxes".

The adjoint boundary conditions are found by equaling integrals over Γ_i to zero in (10),(12) and (13):

$$\left(\mathbf{A}_n - \mathbf{A}_{vn}\right)^T\psi + \mathbf{C}_n^T\partial_x\psi + \mathbf{E}_n^T\partial_y\psi + \mu_i^T\partial_{\mathbf{s}}\mathbf{G}_i + \partial_{\mathbf{s}}\Phi = 0 \quad (17)$$
$$\partial_{\mathbf{q}_x}\Phi + \mu_i^T\partial_{\mathbf{q}_x}\mathbf{G}_i = \tilde{\mathbf{C}}_n^T\psi \quad (18)$$
$$\partial_{\mathbf{q}_y}\Phi + \mu_i^T\partial_{\mathbf{q}_y}\mathbf{G}_i = \tilde{\mathbf{D}}_n^T\psi \quad (19)$$

with:

$$\mathbf{A}_n = \mathbf{A}n_x + \mathbf{B}n_y \quad \mathbf{A}_{vn} = \mathbf{A}_v n_x + \mathbf{B}_v n_y$$
$$\mathbf{C}_n = \mathbf{C}n_x + \mathbf{D}n_y \quad \mathbf{E}_n = \mathbf{E}n_x + \mathbf{F}n_y$$
$$\tilde{\mathbf{C}}_n = \mathbf{C}n_x + \mathbf{E}n_y \quad \tilde{\mathbf{D}}_n = \mathbf{D}n_x + \mathbf{F}n_y$$

2.2 Convective variation

Let us first recall that the *convective* variation (i.e. the variation due to the movement of the domain) of a functional of the form:

$$\mathcal{J} = \int_\Gamma f\, d\Gamma$$

is:

$$\delta\mathcal{J}^{\text{conv}} = \int_\Gamma\left(\vec{\nabla}f\cdot\vec{\omega} + \kappa f\right)\, d\Gamma \quad (13)$$

with:

$$\kappa = \frac{\partial_\sigma\vec{\omega}\cdot\vec{t}}{\left\|\vec{t}\right\|^2}$$

where:

- the boundary Γ is described by the parameter σ: $\Gamma : (x(\sigma), y(\sigma))$;
- $\vec{t}$ is the tangent vector: $\vec{t} = (\, d_\sigma x\; d_\sigma y\,)$;
- $\vec{\omega}$ is the boundary displacement due to the change of the design variables.

For a functional of the form $\mathcal{J} = \int_\Omega f\, d\Omega$, the convective variation is:

$$\delta\mathcal{J}^{\text{conv}} = \int_\Gamma f\omega_n\, d\Gamma$$

where ω_n is the projection of $\vec{\omega}$ to the normal to the surface Γ

Applying the formula above to the Lagrangian of the system (6) leads to the following convective variation (where Gauss' formula is applied):

$$\delta\mathcal{L}^{\text{conv}} = \int_{\Gamma_f}\vec{\nabla}(\mathcal{H} + \psi^T\mathbf{f}_n)\cdot\vec{\omega} + (\mathcal{H} + \psi^T\mathbf{f}_n)\kappa\, d\Gamma_f + \int_{\Gamma_f}\left[\left(\mathbf{f} - \mathbf{f}_v\right)^T\partial_x\psi + \left(\mathbf{g} - \mathbf{g}_v\right)^T\partial_y\psi\right]\omega_n\, d\Gamma \quad (20)$$

If the adjoint equations hold, the variation of the Lagrangian is formed by the convective variation (20) and the local variation due to the design equations (8). Considering that $\vec{\omega} = \vec{\xi}\delta\mathbf{u}$, the optimality condition is:

$$\delta\mathcal{L} = \mathcal{G}\delta\mathbf{u} \quad (21)$$

where:

$$\delta\mathcal{G} = \int_{\Gamma_f}\left[\frac{\partial\mathcal{H}_f}{\partial\mathbf{u}} + \left(\Phi + \psi^T\mathbf{f}_n\right)\kappa + \vec{\nabla}\left(\Phi + \psi^T\mathbf{f}_n\right)\cdot\vec{\xi} + \mu_f^T\vec{\nabla}\mathbf{G}\cdot\vec{\xi} + \left[\left(\mathbf{f} - \mathbf{f}_v\right)^T\partial_x\psi + \left(\mathbf{g} - \mathbf{g}_v\right)^T\partial_y\psi\right]\xi_n\, d\Gamma_f\right] \quad (22)$$

This equation is the *design* equation. The quantity $\mathcal{G}$ can be used as a gradient in an optimization procedure.

In summary, the optimality condition has the form of three sets of equations:

- the *state* equations (2) and boundary conditions (3);
- the *adjoint* equations (16) and boundary conditions (17,18,19);
- the *design* equations (22).

3 Application to two dimensional Navier-Stokes equations

3.1 The Navier-Stokes equations

The 2D Navier-Stokes Equations have the form:

$$\partial_x(\mathbf{f}-\mathbf{f}_v)+\partial_y(\mathbf{g}-\mathbf{g}_v)=0 \qquad (23)$$

where $\mathbf{f},\mathbf{g}$ are the inviscid fluxes , and the viscous fluxes are given by:

- $\mathbf{f}_v=[0,\tau_{xx},\tau_{xy},u\tau_{xx}+v\tau_{xy}-h_x]^T$
- $\mathbf{g}_v=[0,\tau_{xy},\tau_{yy},u\tau_{xy}+v\tau_{yy}-h_y]^T$

The components of the stress tensor τ and the thermal flux $\vec{q}$ can be expressed in term of the first derivative of the primitive variables $\mathbf{q}\equiv[\rho,u,v,T]$:

- $\tau_{xx}=\frac{4}{3}\mu\partial_x u-\frac{2}{3}\mu\partial_y v$
- $\tau_{yy}=\frac{4}{3}\mu\partial_y v-\frac{2}{3}\mu\partial_x u$
- $\tau_{xy}=\mu\left(\partial_y u+\partial_x v\right)$
- $\vec{h}=-k\vec{\nabla}T$

with $\mu(T)$ the dynamic viscosity, and $k(T)$ the thermal conductivity.

The far-field boundary conditions are the same as for the Euler equations. The conditions for a wall are:

- the no-slip condition: $u=v=0$
- the isothermal condition $T=T_0$ or adiabatic condition $\vec{\nabla}T\cdot\vec{n}=0$

3.2 Optimization problem

The functional to minimize has the form:

$$\mathcal{J}=\int_{\Gamma_f}\Phi(p,\tau_w,\tau_n)d\Gamma_f \qquad (24)$$

with p the pressure and τ_w the wall shear stress. The wall normal viscous stress τ_n, which is zero, has to be introduced for compatibility reasons in the adjoint boundary conditions [10][2].

3.3 Adjoint equations

The adjoint equations are given by (16). Expanding this equation leads to:

$$\mathbf{A}^T\partial_x\psi+\mathbf{B}^T\partial_y\psi-\mathbf{M}^T\left(\partial_x\tilde{\mathbf{f}}_v+\partial_y\tilde{\mathbf{g}}_v\right)=\mathbf{h}_v \qquad (25)$$

with:

- $\tilde{\mathbf{f}}_v=[0,\,\Gamma_{xx},\,\Gamma_{xy},\,k\partial_x\psi_4]^T$
- $\tilde{\mathbf{g}}_v=[0,\,\Gamma_{xy},\,\Gamma_{yy},\,k\partial_y\psi_4]^T$
- $\mathbf{h}_v=\mathbf{A}_v^T\partial_x\psi+\mathbf{B}_v^T\partial_y\psi$, or in extenso:

$$\begin{aligned}
\mathbf{h}_{v1} &= 0\\
\mathbf{h}_{v2} &= \tau_{xx}\partial_x\psi_4+\tau_{xy}\partial_y\psi_4\\
\mathbf{h}_{v3} &= \tau_{xy}\partial_x\psi_4+\tau_{yy}\partial_y\psi_4\\
\mathbf{h}_{v4} &= k_T\left(\vec{\nabla}\psi_4\cdot\vec{\nabla}T\right)\\
&+ \frac{\mu_T}{\mu}\left(\tau_{xx}\partial_x\psi_2+\tau_{xy}\partial_x\psi_3+\tau_{xy}\partial_y\psi_2+\tau_{yy}\partial_y\psi_3\right)
\end{aligned}$$

where μ_T, k_T are the derivatives of the viscosity and conductivity with respect to the temperature. The components of the adjoint tensor Γ are:

- $\Gamma_{xx}=\frac{4}{3}\mu\left(\partial_x\psi_2+u\partial_x\psi_4\right)-\frac{2}{3}\left(\partial_y\psi_3+v\partial_y\psi_4\right)$
- $\Gamma_{yy}=\frac{4}{3}\mu\left(\partial_y\psi_3+v\partial_y\psi_4\right)-\frac{2}{3}\left(\partial_x\psi_2+u\partial_x\psi_4\right)$
- $\Gamma_{xy}=\mu\left(\partial_x\psi_3+\partial_y\psi_2+u\partial_y\psi_4+v\partial_x\psi_4\right)$

3.4 Adjoint boundary conditions on a wall

The adjoint boundary conditions for a non-slip wall ar computed by the equations (17,18,19). A lenghty calculation gives the following result:

- for ψ:
  - $\psi_s = \Phi_{\tau_w}$
  - $\psi_n = \Phi_{\tau_n} = -\Phi_p$
  - $\psi_4 = 0$ (isothermal BC), **or** $k\vec{n}\cdot\vec{\nabla}\psi_4 = \Phi_{\tau_w}\tau_w\left(1-\frac{\mu_T}{\mu}\right)$ (adiabatic BC).
- for μ:
  - $\mu_n = -(\rho\psi_1 + \Gamma_n)$
  - $\mu_s = \tau_w\frac{\mu_T}{\mu} - \Gamma_s$
  - $\mu_3 = \Phi_{\tau_w}\tau_w\left(1-\frac{\mu_T}{\mu}\right) - k\vec{n}\cdot\vec{\nabla}\psi_4$ (isothermal BC), **or** $\mu_3 = k\psi_4$ (adiabatic BC).

where (ψ_n, ψ_s) and (μ_n, μ_s) are the normal and tagential to boundary components of the vector (ψ_2, ψ_3) and (μ_1, μ_2); Γ_n and Γ_s are the normal and tangential components of the adjoint tensor Γ.

The adjoint boundary conditions also give a *compatibility condition* on the objective function Φ:

$$\Phi_{\tau_n} = -\Phi_p \tag{26}$$

3.5 Design equation

The design equation is calculated using (22). Several terms in $\vec{\nabla}(\Phi + \psi^T\mathbf{f}_n)$ cancel each other due to adjoint boundary conditions, and the gradient is written:

$$\begin{aligned}\mathcal{G} = &\int_{\Gamma_f}\frac{\partial}{\partial\mathbf{u}}(\Phi + p\psi_n - \tau_w\psi_s)\,d\Gamma_f \\ &+\int_{\Gamma_f}(\Phi + p\psi_n - \tau_w\psi_s)\,\kappa\,d\Gamma_f \\ &+\int_{\Gamma_f} p\vec{\nabla}\psi_n\cdot\vec{\xi} - \tau_w\vec{\nabla}\psi_s\cdot\vec{\xi}\,d\Gamma_f \\ &+\int_{\Gamma_f}\mu_n'\vec{\nabla}u_n\cdot\vec{\xi} + \mu_s'\vec{\nabla}u_s\cdot\vec{\xi}\,d\Gamma_f \\ +\int_{\Gamma_f}&\left(p\vec{\nabla}\cdot\vec{\psi} - \frac{\tau_w}{\mu}\Gamma_s - k\vec{\nabla}T\cdot\vec{\nabla}\psi_4\right)\xi_n\,d\Gamma_f\end{aligned} \tag{27}$$

with:

- $\mu_s' = \psi_4\tau_w\left(1-\frac{\mu_T}{\mu}\right) - \Gamma_s$
- $\mu_n' = \rho H\psi_4 - \Gamma_n$

4 Discretization and numerical solution

The optimality condition is composed of three equations (state,adjoint and design). The most common method, which is the one used in this work, is to stand on the intersection of the state and adjoint surfaces, i.e. to have state and adjoint variables that satisfy state and adjoint equations, and to march to the design surface . This method is the well-known descent method, commonly used in mechanical optimization.

4.1 Navier-Stokes equations

The Navier-Stokes equations (23) are discretized by finite-volume method and solved by an implicit time marching method[9] .

Integrating the unsteady equations on a finite volume Ω_i and using the Gauss theorem yields:

$$\partial_t\int_{\Omega_i}\mathbf{s}\,d\Omega_i + \oint_\Gamma(\mathbf{F}(\mathbf{s},\mathbf{n}) - \mathbf{F}_v(\mathbf{s},\mathbf{n}))\,d\Gamma \tag{28}$$

where $\mathbf{F} = \mathbf{f}n_x + \mathbf{g}n_y$ and $\mathbf{F}_v = \mathbf{f}_v n_x + \mathbf{g}_v n_y$ are respectively the advective and viscous fluxes through the boundary Γ of the cell.

Considering polygonal cells, and integrating the flux using an n_g-points Gauss formula, gives:

$$\partial_t\int_{\Omega_i}\mathbf{s}\,d\Omega_i + \sum_{j\in N_i} l_j\sum^{n_g} w_g\left(\tilde{\mathbf{F}}_j - \tilde{\mathbf{F}}_{vj}\right) = 0 \tag{29}$$

where l_j is the length of the edge j, w_g the weight of the Gauss point,and N_i is the set of edges defining the control volume Ω_i. The numerical inviscid flux are computed using the Roe flux-difference splitting:

$$\tilde{\mathbf{F}} = \frac{1}{2}\left[\mathbf{F}(\mathbf{s}^L,\mathbf{n}) + \mathbf{F}(\mathbf{s}^R,\mathbf{n}) - |\tilde{\Omega}(\mathbf{s}^L,\mathbf{s}^R,\mathbf{n})|\,(\mathbf{s}^R - \mathbf{s}^L)\right] \tag{30}$$

where $\tilde{\Omega}$ is the Jacobian matrix evaluated using Roe's average of $\mathbf{s}^R, \mathbf{s}^L$, the latter being the value at the right or the left of the edges extrapolated from the center of the adjacent cells. This reconstruction can be of order 0 (constant), 1 (linear) or 2 (quadratic), and may imply complex detector-limiter to deal with flow discontinuities (shocks or slip lines).

A centered discretization is used to discretize the viscous terms of the Navier-Stokes equations. The viscous flux is evaluated at one point located at the middle edge.

Once the spatial part is discretized, the equation is integrated in time with a fully implicit Newton-like method. The following non-linear equation is solved at each time step:

$$\mathcal{F}(\mathbf{s}^{n+1}, \mathbf{s}^n) = \frac{\mathbf{s}^{n+1} - \mathbf{s}^n}{\Delta t} + \mathcal{R}(\mathbf{s}^{n+1}) = 0 \quad (31)$$

If this equation is linearized, the following linear system is obtained:

$$\left(\frac{\mathbf{I}}{\Delta t} + \mathcal{J}\right) \mathbf{s}^{n+1} = -\mathcal{F}(\mathbf{s}^n, \mathbf{s}^n) \quad (32)$$

with $\mathcal{J} = \frac{\partial \mathcal{R}}{\partial \mathbf{s}}$, the jacobian matrix. This system is solved by a 1-step Newton method using a ILU0 preconditioned GMRES Algorithm[14].

4.2 Adjoint equations

The continuous adjoint equations (16) are discretized with the same finite-volume method as the Euler equations: integrating (16) over a cell i gives (using Gauss theorem):

$$\partial_t \int_{\Omega_i} \psi \, d\Omega_i + \int_{\Omega_i} \mathbf{A}^T \partial_x \psi + \mathbf{B}^T \partial_y \psi \, d\Omega_i + \int_{\Omega_i} \partial_x \tilde{\mathbf{f}} + \partial_y \tilde{\mathbf{g}} \, d\Gamma = \int_{\Omega_i} \mathbf{h} \, d\Omega_i \quad (33)$$

or, using cell averaged values and applying Gauss theorem on integrals containing $\partial_x \psi, \partial_y \psi$,

$$\partial_t \psi_i + \sum_{j \in N_i} l_j \sum^{n_g} w_g \left(\tilde{\mathbf{G}}_{ij} + \mathbf{M}_i^T \tilde{\mathbf{F}}_{vn}(\psi_{ij})\right) = \mathbf{A}_{vn}^T \sum_{j \in N_i} l_j \sum^{n_g} w_g \psi_{ij} \quad (34)$$

where the numerical adjoint advective flux $\tilde{\mathbf{G}}_{ij}$ is computed with a Roe-like flux splitting formula:

$$\tilde{\mathbf{G}}_{ij} = \frac{1}{2}\left[\mathbf{A}_n^T(\mathbf{s}_i)\left(\psi^L + \psi^R\right) - \nu\left(\psi^R - \psi^L\right)\right] \quad (35)$$

with $\mathbf{A}_n = \mathbf{A}n_x + \mathbf{B}n_y$ and ν an artificial dissipation factor. This dissipation factor is mandatory for the convergence of adjoint equations used in transsonic flows problems. It will be shown that in such problems, the discontinuous state variables imply Dirac peaks in the adjoint solution. A lot of dissipation is then needed to avoid divergence problems.

As for the state variables, the adjoint variables on the edges are extrapolated from the nodal values using a constant, linear, or quadratic reconstruction. The adjoint viscous flux $\tilde{\mathbf{F}}_v$ is evaluated in a similar manner as the viscous flux of the Navier-Stokes equations.

Time derivatives are added to the adjoint equations to obtain a time dependent system of equations, which is then solved iteratively by using the same method as for the Navier-Stokes equations, i.e. fully-implicit Newton-like integration.

4.3 Optimization procedure

The optimization procedure presented here consists in transforming the highly non-linear shape optimization problem:

$$\begin{aligned} &\min && f(\mathbf{x}) \\ &\text{with} && c_i(\mathbf{x}) \leq \bar{c}_i \quad \text{for } i = 1 \cdots n_c \end{aligned} \quad (36)$$

where the function and constraints are not explicitely known into a sequence of simpler (explicit) problems. At each iteration k, a subproblem P^k :

$$\begin{aligned} &\min && \tilde{f}^{(k)}(\mathbf{x}) \\ &\text{with} && \tilde{c}_i^{(k)}(\mathbf{x}) \leq \bar{c}_i \quad \text{for } i = 1 \cdots n_c \end{aligned} \quad (37)$$

is constructed and solved by the CONLIN procedure developed at the University of Liège[6].

In this procedure, the key point consists in finding a good modelling of the objective function and constraints. Two different approaches were tested.

The first method is called "Method of Moving Asymptotes" (MMA)[11]. At iteration k, a function f is approximated by:

$$\tilde{f}^{(k)}(\mathbf{x}) = \sum_{i=1}^{n} \frac{p_i}{U_i^{(k)} - x_i} + \frac{q_i}{x_i - L_i^{(k)}} + r \quad (38)$$

The parameters r,p_i and q_i are adjusted such that :

$$\begin{aligned} \tilde{f}(\mathbf{x}^{(k)}) &= f(\mathbf{x}^{(k)}) \\ \left.\frac{\partial \tilde{f}}{\partial x_i}\right|_{x^{(k)}} &= \left.\frac{\partial f}{\partial x_i}\right|_{x^{(k)}} \end{aligned}$$

In addition p_i is set to zero when $\partial_{x_i} f < 0$ at $\mathbf{x}^{(k)}$ and q_i is set to zero when $\partial_{x_i} f > 0$, such that $\tilde{f}$ is a monotoneous increasing (resp. decreasing) function of x_i:

$$\begin{aligned} p_i^{(k)} &= \max\left[0\,,\,\left(U_i^{(k)} - x_i^{(k)}\right)^2 \left.\frac{\partial f}{\partial x_i}\right|_{x^{(k)}}\right] \\ q_i^{(k)} &= \min\left[0\,,\,\left(x_i^{(k)} - L_i^{(k)}\right)^2 \left.\frac{\partial f}{\partial x_i}\right|_{x^{(k)}}\right] \end{aligned}$$

The parameters $U_i^{(k)}$ and $L_i^{(k)}$ are the asymptotes which are allowed to move from one iteration to another, in order to narrow or broaden the feasible space

The second method of approximation is the "Diagonal Quadratic" method [4]. At iteration k, a function f is modelled by:

$$\tilde{f}^{(k)}(\mathbf{x}) = f(\mathbf{x}^{(k)}) + \sum_{i=1}^{n} b_i (x_i - x_i^{(k)}) + \frac{1}{2} a_i \left(x_i - x_i^{(k)}\right)^2 \tag{39}$$

with:

$$\begin{aligned} b_i &= \left.\frac{\partial f}{\partial x_i}\right|_{x^{(k)}} \\ a_i &= \left.\frac{\partial^2 f}{\partial x_i^2}\right|_{x^{(k)}} \end{aligned}$$

As the second derivatives of the function is usually not available, the term a_i has to be approximated. We have employed a rank-1 update procedure for the coefficients a_i:

$$a_i^{(k+1)} = a_i^{(k)} + \alpha \frac{v_i^2}{z} \tag{40}$$

with $\mathbf{v} = \vec{\nabla} f^{(k+1)} - \vec{\nabla} f^{(k)}$, $z = \mathbf{v}^T \left(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\right)$ and α a relaxation factor:

$$\alpha = \min\left(1\,,\,\frac{\sqrt{\sum_i a_i^2}}{z||\mathbf{v}||}\right)$$

5 Geometry considerations

5.1 Airfoil parametrization

The profile is defined by the formula:

$$y(x) = y_c(x) \pm y_t(x) \tag{41}$$

where y_t and y_c are the thickness and camber distributions defined as follow (see figures 1(a) and 1(b)):

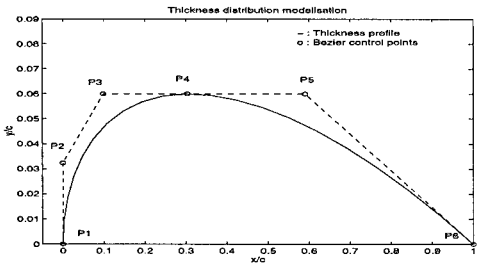
- the thickness distribution is modelled by two Bezier curves (one curve of degree 3 on the front part and one of degree 2 on the aft part); the design variables are the position and value of maximum thickness, the leading edge radius and the trailing edge wedge angle.
- the camber distribution is modelled by two Bezier curves of order 2; the design variables are the position and value of maximum camber, and the first derivative at leading and trailing edges.

The airfoil section is then defined by eight parameters.

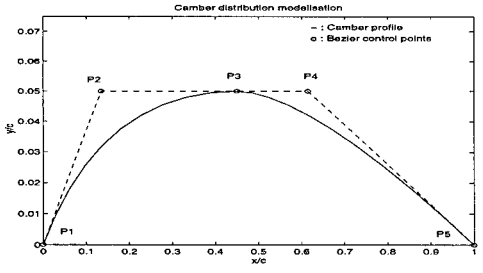
5.2 Mesh movement

During the optimization cycle, the geometry is modified and the computational mesh must be adapted. A straightforward method consists in generating a new mesh at each iteration, but this requires a fast and robust mesh generation technique. In the method used here, which is called "spring analogy", the vertices of the mesh are moved, without changing the connectivity nor the number of mesh points. The mesh is considered as a spring network that reacts to a perturbation of the boundary by adjusting the vertex position in such a way that the deformation energy is minimal. This leads to a system of non-linear equations that are solved by a Newton algorithm.

The spring analogy method is well suited for unstructured meshes for inviscid flows but needs some modifications to be used on hybrid meshes for viscous flows, because of the high aspect ratio of the cells in the boundary layer that can produce overlapping or highly distorted meshes. A vertex i near



(a) Thickness distribution



(b) Camber distribution

Figure 1: Airfoil modelisation

the moving boundary (in the boundary layer) is moved according to the formula:

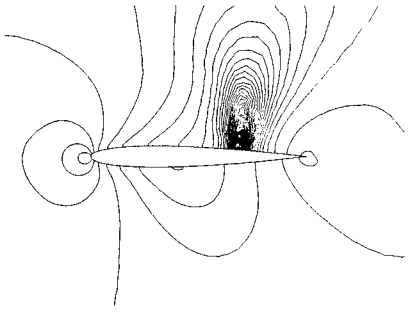
$$\mathbf{x}_i^{\text{new}} - \mathbf{x}_i^{\text{old}} = \alpha(d)\left(\mathbf{x}_j^{\text{new}} - \mathbf{x}_j^{\text{old}}\right) \qquad (42)$$

where j is the nearest point on the boundary with respect to i, and $\alpha(d) \in [0\,1]$ is a decreasing function of the distance to the boundary d, and vanishes for $d > d_{\text{cut}}$. This procedure is applied to all the vertices for which $d < d_{\text{cut}}$, and the remaining vertices are updated with the spring analogy.

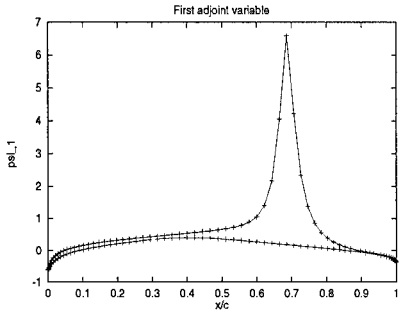
6 Results

6.1 Inviscid transsonic Lift/Drag Ratio optimization

The test case consists in maximizing the lift to drag ratio at M=0.8 and an incidence of 2 degrees, with a lower bound to the maximumthickness of the airfoil (12% of chord). In such a configuration, there is a shock on the lee-side of the airfoil. This discontinuity in the state variable induces Dirac peaks in the adjoint solution The first adjoint variable obtained at the first iteration step is shown on figure 2.



(a) ψ_1 field



(b) ψ_1 on airfoil

Figure 2: First adjoint variable

This peak is smoothed by the numerical computation which involves a numerical dissipation scheme. Attempts to use a less diffusive flux (i.e. a Roe-type flux difference splitting) may lead to a failure of convergence.

This type of discontinuity is inherent to the problem, because the shock in the flow solution is not differentiable. Such Dirac discontinuities can be found also when using the flow sensitivity derivatives to compute the gradient[8].

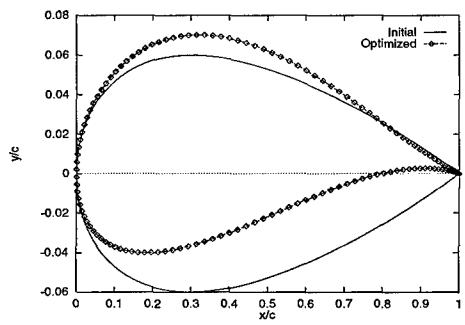
The problem of minimizing the ratio $\frac{C_d}{C_l}$ is not obvious because the cost function cannot be written in the form (24), although C_l and C_d have this form.

The gradient of $f = \frac{C_d}{C_l}$ is

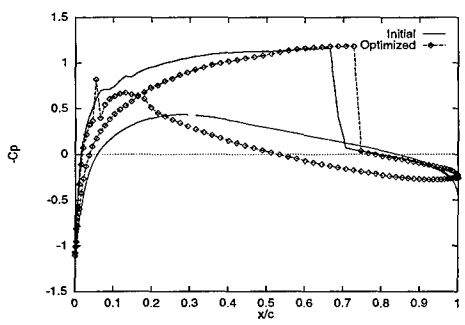
$$\nabla f = \frac{1}{C_l}\left(\nabla C_d - f \nabla C_l\right) \qquad (43)$$

One straightforward approach is to compute *two* adjoint problems, with functions C_l and C_d and to put their gradients in the formula above. The method used here consists in using the functional $\tilde{f} = \frac{1}{C_{l0}}(C_d - f_0 C_l)$, defined at the current design point α_0, for the computation of the gradient. This function has the same gradient as the initial cost function at the current design point, and has the form (24), because it is linear in C_l and C_d. Its gradient only requires *one* adjoint computation.

After twenty iterations, the lift to drag ratio has increased from $2.03 \cdot 10^{-2}$ to 10.65. The final airfoil is slightly thinner than initial airfoil, with camber and a sharper trailing edge, see figure 3. The lee-side shock has slightly increased and moved backward (fig. 3).



(a) Shape modification



(b) Cp comparison

Figure 3: Shape and Cp comparison

6.2 Lift maximization

The first test case is a lift coefficient maximization of an airfoil in subsonic viscous flow (Mach=0.5, Re=100.000). The initial airfoil is a slightly cambered NACA0012, and an upper bound on the drag coefficient is imposed. This problem was solved with different approximations for the objective and the constraint (in the text below, MMA/QUA stands for "Method of Moving asymptots for the objective and diagonal quadratic method for the constraint", for example). Results are presented in table 1 and 2.

Method	C_L	C_D	$\frac{C_L}{C_D}$
Initial	$1.778 \cdot 10^{-1}$	$3.024 \cdot 10^{-2}$	5.88
MMA/MMA	$2.399 \cdot 10^{-1}$	$2.771 \cdot 10^{-2}$	8.66
MMA/QUA	$2.449 \cdot 10^{-1}$	$2.774 \cdot 10^{-2}$	8.83
QUA/QUA	$2.273 \cdot 10^{-1}$	$2.945 \cdot 10^{-2}$	7.72

Table 1: Aerodynamic coefficients

Method	ΔC_L	ΔC_D	$\Delta \frac{C_L}{C_D}$
MMA/MMA	+34.9%	−8.30%	+47.2%
MMA/QUA	+37.7%	−8.30%	+50.1%
QUA/QUA	+27.8%	−2.60%	+31.2%

Table 2: Performance of the different approximation techniques

It can be noticed that the all-MMA or MMA/QUA strategies work better than the all-QUA strategy. That can be due to two facts:

- the MMA approximation is well suited for monotonically decreasing or increasing functions (and that can be the case for C_L);
- the QUA method relies on the quasi-Newton approximation of the Hessian matrix, which can be quite inaccurate, especially during the first iterations.

Figure 4 shows the initial and optimized shapes for MMA/QUA and QUA/QUA computations (MMA/MMA gives almost the same airfoil as MMA/QUA).

Figure 5 shows the Mach field around the optimized airfoil (MMA/QUA).

The optimized airfoil is cambered (1.1%) to increase lift, with the position of maximum camber displaced

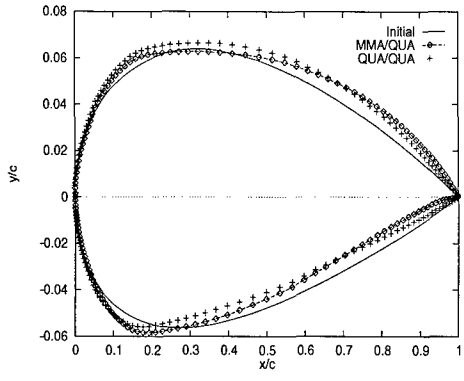


Figure 4: Initial and optimized shapes

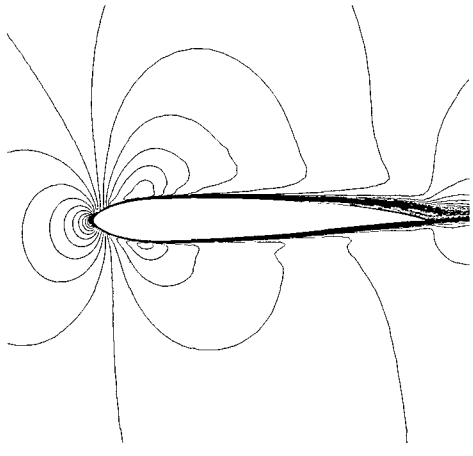


Figure 5: Mach field around optimized airfoil (Mach max.=0.89253)

toward aft (at 75% of chord), in such a way that the recirculation bubble is smaller (see figure 6), thus leading to a decrease in the viscous drag coefficient.

6.3 Inverse design

Inverse design problems can be treated as optimization problems, the function to optimize being the least square distance between a target pressure field and the actual pressure. Although optimization techniques are much more expensive in term of calculation time, they always succeed in finding an optimum value where inverse design techniques can fail. Inverse design problems are also useful when testing algorithms, because the optimal solution is known.

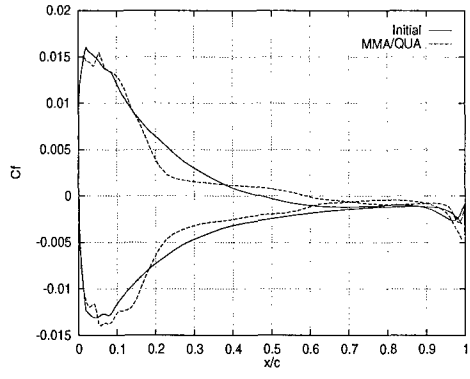


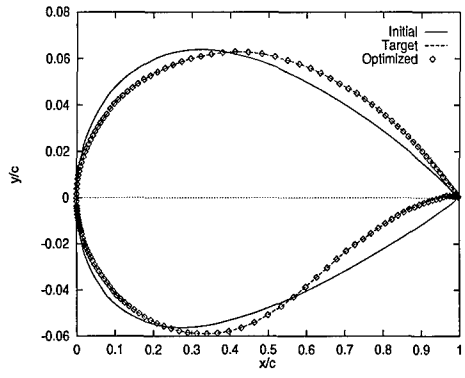
Figure 6: Skin friction coefficient on initial and optimized airfoils

In the present test case, the pressure of an airfoil (Mach=0.5, Re=100.000) has to be matched to the pressure field of an RAE2822 airfoil in the same flow conditions. The initial airfoil is the same NACA0012 as in the previous test case. The optimization technique used is MMA.

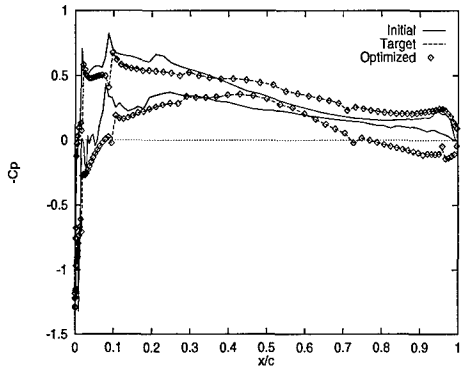
As can be seen on figure 7, the final solution is very close to the target, in term of shape and pressure distribution.

The residual history gives valuables observations of the behaviour of the MMA algorithm: the initial oscillations of the objective functions, due to oscillations of the variable describing camber, are progressively damped by bringing the moving asymptots closer to each other. However this leads to a non-optimal solution; this difficulty was overcome by restarting the algorithm after iteration 13 (see figure 8).

In the previous calculation, the upper and lower bound on the design variables are chosen such that the optimal solution lies on the boundary of the design space. In such a case, the MMA approxiamtion is very efficient, because the objective function is generally a monotonous function. When the solution is inside the design space, MMA fails to find the real optimum because it cannot model functions with one (or more) extremum. Figure 9 shows the result of the same test case, with a larger design space: the computed solution lies on the design space boundary and is not the real optimum.



(a) Shape comparison



(b) Pressure coefficient comparison

Figure 7: Initial, target and optimized shape and Cp comparison

7 Conclusions and prospects

In this paper, we have showed that the adjoint method can give accurate gradient to aerodynamic coefficients. This method, together with fast flow analysis algorithms and a robust constrained optimization procedure, is mandatory for the resolution of aerodynamic shape optimization problems in 2D on small workstations (all the calculations were done on a single CPU HP-C160 workstation).

Further work is needed on the resolution of the adjoint equations, especially to derive the adjoint equations for turbulent flows. Another topic of interest is the study of approximations of the functions; a hybrid method involving both MMA and QUA could be very efficient[12] if a good approximation of the Hessian matrices could be found. Further studies on Hessian matrices[3] could thus be valuable.

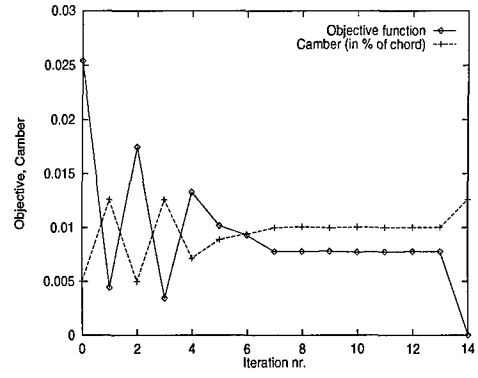


Figure 8: History of objective function and camber

8 Acknowledgment

The first author gratefully acknowledges the full financial support provided by th Belgian National Fund for Scientific Research (F.N.R.S.)

References

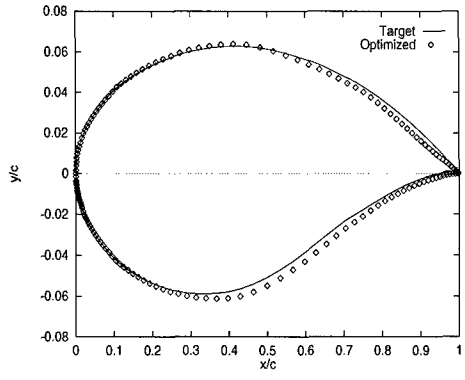
[1] J. Reuther A. Jameson. Control Theory Based Airfoil Design using the Euler Equations. AIAA Report 94-4272-CP, American Institute for Aeronautics and Astronautics, 1994.

[2] M.D. Salas E. Arian. Admitting the Inadmissible: Adjoint Formulation for Incomplete Cost Functionals in Aerodynamic Optimization. ICASE Report 97-69, NASA Langley Research Center, Hampton, Virginia, 1997.

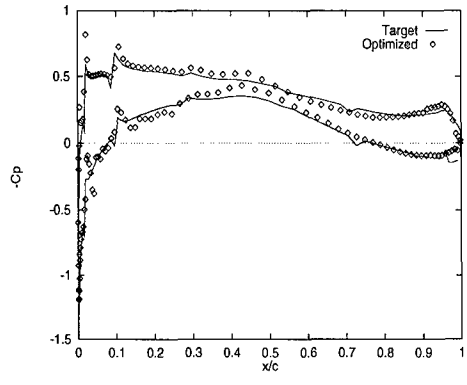
[3] S. Ta'asan E. Arian. Analysis of the Hessian for Aerodynamic Optimization: Inviscid Flow. ICASE Report 96-28, NASA Langley Research Center, Hampton, Virginia, 1996.

[4] C. Fleury. Efficient approximation concepts using second order information. In *AIAA Structural Dynamics and Materials Conference, Williamsburg, USA*, April 18 – 20 1988.

[5] O. Baysal G.W. Burgreen. Three-Dimensional Aerodynamic Shape Optimization Using Dis-



(a) Shape comparison



(b) Pressure coefficient comparison

Figure 9: Target and final shape and Cp when widening the design space

crete Sensitivity Analysis. *AIAA Journal*, 34(9), September 1996.

[6] L.A. Schmit H. Smaoui, C. Fleury. Advances in Dual Algorithms and Convex Approximation Methods. In *AIAA/ASME/ASCE/AHS 29th Structures, Structural Dynamics and Materials Conference, Williamsburg, VI*, 1988.

[7] G.W. Burgreen J.C. Newmann III, A.C. Taylor. An Unstrucured Grid Approach to Sensitivity Analysis and Shape Optimization Using the Euler Equations. In *12th AIAA Computational Fluid Dynamics Conference, San Diego, California*, June 1995.

[8] M.D. Gunzburger J.R. Appel. Difficulties in Sensitivity Calculations for Flows with Discontinuities. *AIAA Journal*, 35(5), May 1997.

[9] J.A. Essers P. Rogiest M. Delanaye, P. Geuzaine. A Second Order Finite Vlume Scheme solving Euler and Navier-Stokes Equations on Unstructured Adaptive Grids with an Implicit Acceleration Procedure. In *77th AGARD FDP Symposium on Progress and Challenges in CFD methods and algorithms, Sevilla, Spain*, 1995.

[10] B. Soemarwoto. The Variational Method for Aerodynamic Optimization Using the Navier-Stokes Equations. ICASE Report 97-71, NASA Langley Research Center, Hampton, Virginia, 1997.

[11] K. Svanberg. The method of moving asymptotes – a new method for structural optimization. *International Journal for Numerical Methods in Engineering*, 24:359 – 373, 1987.

[12] K. Svanberg. Some second order methods for structural optimization. In *Optimization of Large Structural Systems, Vol. I*, pages 567 – 578. Kluwer Academic Publishers, 1993.

[13] V. Venkatakrishnan W. Kyle Anderson. Aerodynamic Shape Optimization on Unstructured Grids with a Continuous Adjoint Formulation. In *35th Aerospace Sciences Meeting and Exhibit, Reno, NV*, January 97.

[14] M. Shultz Y. Saad. G.M.R.E.S.: a generalized minimal residual algorithm for solving nonsymmetric linear system. *SIAM J. Sci. Stat. Comp.*, 7, 1986.